JOURNAL OF COMBINATORIAL THEORY, Series B 36, 161–188 (1984)

Testing Membership in Matroid Polyhedra

WILLIAM H. CUNNINGHAM*

Institut für Operations Research, Universität Bonn, West Germany

Communicated by the Editors

Received November 13, 1981

Given a matroid M on E and a nonnegative real vector $x = (x_j: j \in E)$, a fundamental problem is to determine whether x is in the convex hull P of (incidence vectors of) independent sets of M. An algorithm is described for solving this problem for which the amount of computation is bounded by a polynomial in $|E|$, independently of x, allowing as steps tests of independence in M and additions, subtractions, and comparisons of numbers. In case $x \in P$, the algorithm finds an explicit representation for x which has additional nice properties; in case $x \notin P$ it finds a most-violated inequality of the system defining P. The same technique is applied to the problem of finding a maximum component-sum vector in the intersection of two matroid polyhedra and a box.

1. INTRODUCTION

Let M be a matroid on E and let $x = (x_j: j \in E)$ be a given real-valued vector. The main result of this paper is an efficient combinatorial algorithm to determine whether x is in the convex hull P of (incidence vectors of) independent sets of M. A fundamental theorem of Edmonds [8] gives an explicit description of P. (In this paper r denotes the rank function of M and the sum over those components of a vector y which are indexed by elements of a set S is denoted by $y(S)$. Also, we often use the same symbol to represent a set or its incidence vector. Finally, n denotes the cardinality of E.)

THEOREM 1.1. $P = \{x: x \geqslant 0,\ x(A) \leqslant r(A)\ \textit{for all}\ A \subseteq E\}$.

In the case in which $x \notin P$ our algorithm terminates with a set $A \subseteq E$ for which $x(A) > r(A)$ (or an element $j \in E$ with $x_j < 0$). In the alternative case

* Supported by Sonderforschungsbereich 21 (DFG), Institut für Operations Research, Universität Bonn, West Germany. On leave from Department of Mathematics and Statistics, Carleton University, Ottawa, Canada. Research partially supported by an N.S.E.R.C. of Canada operating grant.

0095-8956/84 $3.00

when $x \in P$, the algorithm terminates with an explicit, economical representation for x as a convex combination of independent sets. Therefore, we obtain an efficient constructive proof of Theorem 1.1.

There is another, earlier, method of answering the question "Is $x \in P$?" in polynomial time. It has been discovered by Grötschel, Lovász and Schrijver [13] and is based on the ellipsoid method of linear programming. This method can be used as well to find a violated inequality in case $x \notin P$, or to find an expression for x in terms of independent sets, in case $x \in P$. Our approach, however, seems to have a number of advantages over this ellipsoid method. First, the only arithmetic steps it uses are additions, subtractions, and comparisons. Second, the number of arithmetic and other elementary steps is bounded by a polynomial in n, *independent of* x; that is, our algorithm is "strongly polynomial." Third, the present method can be used to prove Theorem 1.1 (and stronger results), whereas the method of [13] uses Theorem 1.1 in a fundamental way. Fourth (and this is to some extent a matter of opinion), our method seems to be computationally practicable, and the method of [13] at the present time does not.

Our algorithm actually proves the following slight extension of Theorem 1, by terminating with $y \in P$ and $A \subseteq E$ such that $y \leqslant x$ and $y(E) = r(A) + x(E \setminus A)$. (It is obvious that for any such y and A, $y(E) \leqslant r(A) + x(E \setminus A)$.) This result [8] is the "vector rank" formula for P, considered as a polymatroid.

THEOREM 1.2. *For any* $x = (x_j : j \in E) \geqslant 0$, $\max(y(E) : y \in P,\ y \leqslant x) = \min(r(A) + x(E \setminus A) : A \subseteq E)$.

Any minimizing set A in Theorem 1.2 actually provides a "most violated" inequality in the case when $x \geqslant 0$ and $x \notin P$. This follows because, for any $A \subseteq E$, $r(A) - x(A)$ and $r(A) + x(E \setminus A)$ differ only by the constant $x(E)$.

There are other consequences of the algorithm which imply that, when $x \in P$, an expression for x as a convex combination of independent sets can be required to have additional properties. A standard result of polyhedral theory implies that, where $|E| = n$, x can be expressed as a combination of at most $n + 1$ independent sets. This result says very little, however, about the nature of the coefficients in this expression. On the other hand, Theorem 1.3 places strong discreteness restrictions on these coefficients, while still requiring the expression to be polynomial in size, independent of x.

THEOREM 1.3. *There is a maximizing* y *in Theorem* 1.2 *and an expression* $\sum (\lambda_i I_i : i \in J)$ *for* y *as a convex combination of independent sets, satisfying*:

(i) $|J| \leqslant n^4 + 1$,

(ii) *For each* $i \in J$, λ_i *is an integer combination of elements of* $\{x_j : j \in E\} \cup \{1\}$.

The special case of Theorem 1.3 in which x is rational-valued bears an interesting relationship to the matroid partition theorem [7]. This is described in Section 7; it motivated a result of Schrijver [21] which leads to a substantial improvement of Theorem 1.3 itself.

The algorithm described in this paper seems to be an interesting combination of matroid partitioning [7] and network flow techniques [6, 10]. It constructs the maximizing y of Theorem 1.2 by successive augmentations along shortest paths in an auxiliary diagraph. The efficiency of the algorithm depends on the fact that each augmenting path is found by "consistent breadth-first search," a natural refinement of the usual labelling technique. Such a path has a vertex sequence which is least with respect to a lexicographical ordering; this idea was introduced by Lawler and Martel [15] and Schönsleben [20].

The contents of the paper can be summarized as follows. Section 2 gives a basic augmenting path approach to the membership problem, and proves Theorems 1.1 and 1.2. Section 3 provides the framework for proving that certain classes of augmenting path algorithms require only polynomially many augmentations. Section 4 contains the main result of the paper. It describes a refined version, Algorithm 4.1, of the augmenting path algorithm, and proves, using the theory of Section 3, that it is strongly polynomial. Theorem 1.3 is also proved. Section 5 applies similar techniques to problems on matroid intersection polyhedra. It gives algorithms for finding a maximum component-sum vector in the intersection of two matroid polyhedra with a box, and for testing membership in the blocking and anti-blocking polyhedra associated with the common bases of two matroids. Section 6 describes the problem of submodular function minimization, of which Algorithm 4.1 solves a special case, and extends the algorithm to solve some other problems of this type. The last section consists of a few remarks.

2. The Idea of the Algorithm

In this section we present a few preliminary results, and prove Theorems 1.1 and 1.2. Our matroid terminology mostly conforms with Welsh [22]. Let M be a matroid on E and let $\mathcal{I}$ and r denote its family of independent sets and its rank function, respectively. A minimal nonindependent set is called a *circuit*. We use repeatedly the following elementary result. (In this paper $A+e$, $A-f$, $A+e-f$ often replace the more cumbersome expressions $A \cup \{e\}$, $A \setminus \{f\}$, $(A \cup \{e\}) \setminus \{f\}$.)

LEMMA 2.1. *Let $I \in \mathscr{I}$ and $e \in E$. Then $I + e$ contains at most one circuit of M.*

If $I \in \mathscr{I}$, $e \in E$, and $I + e \notin \mathscr{I}$, we denote by $C(I, e)$ the unique circuit of M contained in $I + e$. (Of course, $e \in C(I, e)$.) In general, use of the notation $C(I, e)$ implies that $I \in \mathscr{I}$ and $I + e \notin \mathscr{I}$.

In order to motivate the approach we shall take, we consider some special choices for x. Suppose first that, for some positive integer k, $x_j = 1/k$ for all $j \in E$. By Theorem 1.1, deciding whether $x \in P$ is equivalent to deciding whether $|A| \leqslant \mathrm{kr}(A)$ for all $A \subseteq E$. Edmonds' matroid partition theorem [7] says that the latter condition holds if and only if there exist k independent sets $(I_i : i \in J)$ whose union is E. The proof is an efficient algorithm to find either $A \subseteq E$ with $|A| > \mathrm{kr}(A)$, or the family $(I_i : i \in J)$ of independent sets. Thus matroid partitioning provides an efficient algorithm for membership-testing in a particular case.

It is useful to recall briefly a version of the partition algorithm. It attempts iteratively to extend a partitioning $(I_i : i \in J)$ of a proper subset of E into k independent sets, to such a partitioning of a larger subset. Initially, we can choose $I_i = \emptyset$, $i \in J$. At each step we form an auxiliary digraph G. The vertex-set of G is E together with a source r and a sink s. There is a directed edge (r, e) for every $e \in E$ for which $e \notin (\bigcup I_i : i \in J)$. There is a directed edge (e, s) for every $e \in E$ for which there is some $i \in J$ with $e \notin I_i$ and $I_i + e \in \mathscr{I}$. Finally, there is a directed edge (e, f) for $e, f \in E$ such that there is some $i \in J$ with $f \in C(I_i, e)$. One can prove an "augmenting path" result, that $\bigcup (I_i : i \in J)$ is a maximum cardinality partitionable subset of E if and only if there is no (r, s) dipath in G. These ideas provide the basis for the partition algorithm, and similar, more general ideas underly our algorithm.

Let us attempt to extend the above approach, still using matroid partition. We assume this time that x is rational, and thus that there exists an integer k such that $b = kx$ is integer-valued. Again by Theorem 1.1, testing whether $x \in P$ is equivalent to deciding whether $b(A) \leqslant \mathrm{kr}(A)$ for all $A \subseteq E$. One can reduce this to an ordinary matroid partition problem by replacing each $e \in E$ by a set S_e of b_e parallel elements, obtaining a new matroid $M' = (E', \mathscr{I}')$. For each $I' \in \mathscr{I}'$ there is a corresponding set $I = \{e : e \in E, S_e \cap I' \neq \emptyset\} \in \mathscr{I}$, having the same cardinality as I', so the rank function r' of M' satisfies $r'(A) = r(\{e : e \in E, S_e \cap A \neq \emptyset\})$ for all $A \subseteq E'$. We can use matroid partitioning to determine whether $|A'| \leqslant \mathrm{kr}'(A')$ for all $A' \subseteq E'$. For this, it is clearly enough to consider only sets A' of the form $\bigcup (S_e : e \in A)$ for some $A \subseteq E$. For such sets $|A'| = b(A)$ and $r'(A) = r(A)$, so matroid partitioning on M' can be used to determine whether $b(A) \leqslant \mathrm{kr}(A)$ for all $A \subseteq E$, that is, to determine whether $x \in P$.

A straightforward implementation of the above approach leads to an algorithm for which the computation bound grows at least linearly with

$kb(E)$, a distinctly unpleasant prospect. However, some ways to improve this approach are apparent. Since each set $I' \in \mathscr{I}'$ corresponds to some $I \in \mathscr{I}$, a family of k disjoint subsets from $\mathscr{I}'$ can be represented by a family $(I_i: i \in J)$ of subsets from $\mathscr{I}$ together with positive integers a_i, $i \in J$, such that $\sum(a_i: i \in J) = k$ and $\sum(a_i: e \in I_i) \leqslant b_e$ for each $e \in E$. Where $\lambda_i = a_i/k$, this is equivalent to $\lambda_i > 0$, $\sum(\lambda_i: i \in J) = 1$, and $\sum(\lambda_i I_i: i \in J) \leqslant x$, that is, an explicit expression for some $y \leqslant x$ as a convex combination of independent sets of M. This is, in fact, what our algorithm will work with, even when x is not assumed to be rational-valued. (This analysis of the rational case can be carried even further, as pointed out by Bixby [1] and Wolsey [23]. Namely, one can apply a "scaling" approach of the type described by Edmonds and Karp [10] to the integers b_e, $e \in E$, together with matroid partition to obtain a membership algorithm which is polynomial in n and the logarithm of the largest numerator or denominator of the components of x. Of course, such an algorithm is not strongly polynomial, and applies only to the rational case.)

With the above ideas as motivation, we return to the discussion of the general problem. The algorithm we will describe, like the maximum flow algorithm and the matroid partition algorithm, maintains a feasible solution y to the maximization problem which is being solved. At each step it attempts to find an improvement to that solution by finding an augmenting path in a certain auxiliary digraph. When no such path exists, the set of reachable vertices in that digraph determines a solution A to the minimization problem such that y and A give equality in the min–max theorem, so that y and A are optimal.

In contrast to the maximum flow analogy, it is not trivial to verify that the current y is feasible to the maximization problem in Theorem 1.2. (Perhaps the title of the paper suggests this!) But there is a short proof that y is feasible: We know from Theorem 1.1 and polyhedral theory that y is a convex combination of a small number of independent sets of M. Therefore, Theorem 1.2 *is* a good characterization of optimality, but this method of verifying feasibility of y seems to be crucial. Therefore, our algorithm maintains y together with an explicit expression for y as a convex combination of independent sets.

Suppose that we have families $(I_i: i \in J)$ of independent sets and $(\lambda_i: i \in J)$ of positive numbers such that $\sum(\lambda_i: i \in J) = 1$. Let $y = \sum(\lambda_i I_i: i \in J)$. Then $y \in P$. Suppose, in addition, that $y \leqslant x$. We define a digraph G as follows. Where r, s are distinct elements not in E, $V(G) = E \cup \{r, s\}$. There is a directed edge (r, e) for every $e \in E$ satisfying $y_e < x_e$. There is a directed edge (e, f) for every pair of distinct elements $e, f \in E$ such that, for some $i \in J$, $f \in C(I_i, e)$. There is a directed edge (e, s) for every $e \in E$ such that, for some $i \in J$, $e \notin I_i$, and $I_i + e \in \mathscr{I}$. Notice that G can be constructed in time polynomial in n, assuming that $|J|$ is polynomially bounded. (As usual,

we assume the existence of an oracle for testing independence or evaluating rank in M.) Notice also that the algorithm can be started with $y = 0$, $J = \{0\}$, $I_0 = \phi$, and $\lambda_0 = 1$. Now we are able to prove an augmenting path theorem.

THEOREM 2.2 (Augmenting path theorem). *If there is a directed path from r to s in G, then there exists $y' \in P$, $y \leqslant y' \leqslant x$, with $y'(E) > y(E)$. If there is no such path, then there exists a set $A \subseteq E$ such that $y(A) = r(A)$ and $y(E \setminus A) = x(E \setminus A)$.*

Proof. First, suppose that there is no such path. Then we choose $A \subseteq E$ such that no edge of G leaves $A + r$. Clearly $y_e = x_e$ for all $e \in E \setminus A$, since otherwise (r, e) is an edge of G. We claim that for each $i \in J$, $|I_i \cap A| = r(A)$. If not, there exist $i \in J$ and $e \in A \setminus I_i$ with $(I_i \cap A) + e \in \mathscr{I}$. Then there are two possibilities. The first is that $I_i + e \in \mathscr{I}$, so that (e, s) is an edge. The second is that $I_i + e \notin \mathscr{I}$, so that $C(I_i, e)$ exists, but is not contained in A; then there exists $f \in I_i \setminus A$ with $f \in C(I_i, e)$, and so (e, f) is an edge. In either case we have a contradiction, and the claim is proved. Therefore,

$$\begin{aligned} y(A) &= \sum (\lambda_i |I_i \cap A| : i \in J) \\ &= \sum (\lambda_i r(A) : i \in J) \\ &= r(A), \qquad \text{as required.} \end{aligned}$$

Now suppose that there exists a directed path from r to s in G. Then there is a sequence $e(1), \ldots, e(m)$ of distinct elements of E and $i(j) \in J$ for $1 \leqslant j \leqslant m$ (the $i(j)$ need not be distinct) satisfying

$$\begin{gathered} x_{e(1)} > y_{e(1)}; \\ e(j+1) \in C(I_{i(j)}, e(j)), \qquad 1 \leqslant j \leqslant m-1; \\ I_{i(m)} + e(m) \in \mathscr{I}. \end{gathered}$$

For each $i \in J$, let $k_i = |\{j : i = i(j)\}|$, and let $\delta > 0$ be sufficiently small. For each $i \in J$, replace λ_i by $\lambda_i - k_i\delta$. For each j, $1 \leqslant j \leqslant m-1$, introduce a new set $I'_{i(j)} = I_{i(j)} + e(j) - e(j+1)$ with coefficient δ, and introduce $I'_{i(m)} = I_{i(m)} + e(m)$ with coefficient δ. Clearly the new families of independent sets and coefficients are legal, and they define a vector $y' = y + \delta\{e(1)\}$, as required. ∎

It is easy to derive Theorems 1.1 and 1.2 from the augmenting path theorem. Moreover, these proofs contain the essential idea behind the algorithm. However, a number of refinements must be added in order to obtain an efficient algorithm. These are described in the next two sections.

Proof of Theorem 1.2. It is clear that, for any $y \in P$ having $y \leqslant x$ and

any $A \subseteq E$, we have $y(E) \leqslant r(A) + x(E \setminus A)$. Therefore, it will be enough to show that there exist y and A giving equality. Choose $y \in P$ with $y \leqslant x$ and $y(E)$ a maximum. Then since the resulting digraph G cannot yield an augmentation, we deduce from the augmenting path theorem the existence of a set A for which $y(E) = r(A) + x(E \setminus A)$. ∎

Proof of Theorem 1.1. It is clear that any $x \in P$ is nonnegative and satisfies $x(A) \leqslant r(A)$ for all $A \subseteq E$. Now suppose that $x \geqslant 0$ and $x \notin P$. Then $x(E) > \max(y(E): y \leqslant x, y \in P) = \min(r(A) + x(E \setminus A): A \subseteq E)$ by Theorem 1.2. Clearly, any minimizing A must satisfy $x(A) > r(A)$. ∎

3. Consistent Breadth-First Search

The rudimentary algorithm suggested by the proof of the augmenting path theorem will be modified in two important ways. One is to introduce a much more sophisticated augmentation step than the one used there, and the other is to choose the paths to be used for augmentation in a special way. We describe the second modification first, partly because it is easier, and postpone discussion of the augmentation to the next section.

The subject of this section is an important technique for proving efficiency of augmenting path schemes, due independently to Lawler and Martel [15] and to Schönsleben [20]. By separating its discussion from the particular context in which we are applying it, we isolate the properties needed to make it work. This helps to motivate the technical results of the next section. It also serves to facilitate other applications of this technique.

Recall that the Dinic–Edmonds–Karp proof [6, 10] of a polynomial bound for the maximum flow algorithm is based on a refinement of the usual labelling method for finding augmenting paths. Namely, one scans vertices in the order that they are labelled. The resulting path, of course, has as few edges as possible. The method of [15, 20] can be viewed as a further refinement of the same sort. In addition to scanning vertices in the order in which they are labelled, we label vertices (from a vertex being scanned) in an order consistent with a fixed linear order of all the vertices. (Then the vertex sequence of the resulting path is lexicographically least with respect to this order, among all shortest (r, s)-dipaths.) The latter rule would automatically be satisfied, for example, if the graph were represented by an adjacency matrix with scanning performed in the natural way. Therefore, this technique also has the property that it is "so simple that it is likely to be incorporated innocently into a computer implementation" [10]. For convenience, let us call this technique *consistent breath-first search* or CBFS; we call the (unique) (r, s) dipath found by CBFS, the CBFS-*path* in G.

There are only two requirements for efficiency of an augmenting path

scheme based on CBFS. They correspond, in the maximum flow algorithm, to the following facts. First, in every path used for augmentation there is at least one *critical* edge: one which becomes unavailable for use in an augmenting path immediately after the augmentation. Second, the only way an edge *becomes* available for use in an augmenting path is by being used in the opposite direction in the previous augmentation. In moving to the current context, the first property remains the same, but the second must be generalized slightly. In the following result we are, of course, thinking of digraphs G_i as successive auxiliary digraphs for some augmenting path scheme.

THEOREM 3.1. *Let $G_0, G_1, \ldots, G_k$ be a sequence of digraphs, each having vertex-set $E \cup \{r, s\}$, and assume a fixed linear ordering of $E \cup \{s\}$. Let Q_i denote the* CBFS *path in G_i for $0 \leqslant i < k$. Suppose that, for $0 \leqslant i < k$:*

(I) *There is an edge of Q_i which is not an edge of G_{i+1}.*

(II) *If (e,f) is an edge of G_{i+1} but not G_i, then $e, f \in E$ and there exist vertices a, b of Q_i with a preceding b on Q_i such that $a = f$ or (a,f) is an edge of G_i and $b = e$ or (e, b) is an edge of G_i.*

Then, where $|E| = n \geqslant 2$, $k \leqslant n^3$.

In the remainder of this section, we assume the hypotheses of Theorem 3.1. Let us call an edge as in (I) *critical*. Since there is at least one critical edge for each i, $0 \leqslant i < k$, we can derive a bound on k by obtaining bounds on the number of times an edge can be critical. The first step in the proof is the same as that for the maximum flow case; namely, we prove that the sequence of path lengths is monotone increasing. For $0 \leqslant i < k$, and $e, f \in E \cup \{r, s\}$, let $d_i(e,f)$ denote the length of a shortest directed path from e to f in G_i ($d_i(e,f) = \infty$ if none exists). It is convenient in the next proofs to abbreviate G_i to G, G_{i+1} to G', d_i to d, d_{i+1} to d', Q_i to Q.

LEMMA 3.2. *For all $f \in E \cup \{r, s\}$, $d'(r,f) \geqslant d(r,f)$ and $d'(f, s) \geqslant d(f, s)$.*

Proof. Let us suppose that there exists some $f \in E \cup \{r, s\}$ such that $d'(r,f) < d(r,f)$, and choose f so that $d'(r,f)$ is as small as possible. Clearly, $d'(r,f) \geqslant 1$, so we can choose the second-last vertex e of a dipath in G' of length $d'(r,f)$. By the choice of f, $d'(r, e) \geqslant d(r, e)$. Since $d'(r, e) = d'(r,f) - 1$, it must be that (e,f) is not an edge of G. Therefore, we may choose vertices a, b of Q as in (II). Of course, $d(r, a) < d(r, b)$. But

$$\begin{aligned} d(r, a) &\geqslant d(r,f) - 1 \\ &\geqslant d'(r,f) \\ &= d'(r, e) + 1 \\ &\geqslant d(r, e) + 1 \\ &\geqslant d(r, b), \end{aligned}$$

a contradiction. This proves that $d'(r,f) \geqslant d(r,f)$. The proof that $d'(f, s) \geqslant d(f, s)$ is similar. ∎

It follows from Lemma 3.2 that the sequence $G_0, G_1, \ldots, G_k$ can be divided into at most $n + 1$ *stages*, during each of which the value of $d_i(r, s)$ remains the same. For each G_i, we let H_i denote the subgraph of G_i consisting of the edges and vertices of the (r, s) dipaths in G_i of length $d_i(r, s)$. (Again, we abbreviate H_i to H, H_{i+1} to H'.) In the special case of maximum flow problems no edge can enter H during a stage. In the current more general context an edge *can* enter H, but only under very restrictive conditions.

LEMMA 3.3. *Suppose that $d(r, s) = d'(r, s)$, and (e,f) is an edge of H' but not of H. Then there exists an edge (a, b) of Q such that (a,f), (e, b) are edges of H.*

Proof. Let $d(r, s) = m + 1$. Then, by Lemma 3.2 $d(r, e) + d(f, s) \leqslant d'(r, e) + d'(f, s) = m$. Hence (e,f) is not an edge of G, so we may choose a, b as in (II). Therefore, $d(r, a) + d(b, s) \leqslant m$, $d(r, e) + d(b, s) \geqslant m$, and $d(r, a) + d(f, s) \geqslant m$. It follows that $d(r, e) + d(f, s) \geqslant m$, and so $d(r, e) + d(f, s) = m$. Therefore all of the above inequalities hold with equality, so $d(r, a) = d(r, e)$, $d(f, s) = d(b, s)$, and (a, b) is an edge of Q. Moreover, $e \neq b$, $a \neq f$, because otherwise $d(r, s) \leqslant m$. Therefore (a,f), (e, b) are edges of G, and hence are edges of H. ∎

For each $e \in E + r$, let $\pi_i(e)$ denote the least element f of $E + s$ such that (e,f) is an edge of H_i. If none exists, $\pi_i(e) = \infty$. (Again, we abbreviate π_i to π and π_{i+1} to π'.) Clearly, the CBFS path Q has the property that $\pi(e) = f$ for every edge (e,f) of Q.

LEMMA 3.4. *If $d'(r, s) = d(r, s)$, then for every $e \in E + r$, $\pi'(e) \geqslant \pi(e)$.*

Proof. Since $\pi'(e)$ is a minimum over a set, $f = \pi'(e) < \pi(e)$ would imply that (e,f) is an edge of H' but not of H. Thus by Lemma 3.3, we have an edge (a, b) of Q and edges (e, b) and (a,f) of H. But then $f < \pi(e) \leqslant b = \pi(a) \leqslant f$, a contradiction. ∎

LEMMA 3.5. *If (e,f) is a critical edge of G_i, then (e,f) is not an edge of H_j for any $j > i$ for which $d_i(r,s) = d_j(r,s)$.*

Proof. Suppose otherwise, and choose a least j. Then by Lemma 3.3 we have edge (a,b) of Q_{j-1} and edges (a,f), (e,b) of H_{j-1}. By choice of Q_{j-1}, $b < f$, and so $\pi_j(e) \leqslant b < f$. But $\pi_i(e) = f$, so this contradicts Lemma 3.4. ∎

Proof of Theorem 3.1. There are at most $2n+1$ edges incident with r or s which can ever be critical, and by (II) each of them can be critical at most once. There are at most $n^2 - n$ other edges; by Lemma 3.5 each of these can be critical at most once per stage, and clearly cannot be critical when $d_i(r,s) = 1$ or 2. Hence $k \leqslant (2n+1) + (n-1)(n^2-n)$, which is $\leqslant n^3$ for $n \geqslant 2$. ∎

As was pointed out to me by R. E. Bixby, Lemma 3.5 is not really needed to prove Theorem 3.1, although it does provide additional insight into the behaviour of the sequence of G_i. The reason is that the vector $(\pi_i(e): e \in E + r)$ strictly increases in the component-wise order during each stage. It is easy to obtain from this observation an $O(n^2)$ bound on the number of G_i per stage.

4. THE ALGORITHM

The augmentation suggested in the proof of the augmenting path theorem is the simplest one for which the proof works. It obviously lacks one of the properties which are needed for application of the CBFS strategy: After such a "crude" augmentation on path Q, Q may still be a path in the new auxiliary digraph. The main business of this section is to define a stronger augmentation step which does allow the application of the results of the last section.

It is useful to introduce a notion of edge capacity in the auxiliary digraph $G = G(J, \lambda)$. An edge (r,e) of G has capacity $u(r,e) = x_e - y_e$. Where $e \in E$, $f \in E + s$, and $e \neq f$, we define $D(e,f)$ to be $\{i: i \in J,\ e \notin I_i \text{ and } f \in C(I_i, e)\}$ if $f \neq s$ and to be $\{i: i \in J,\ e \notin I_i, \text{ and } I_i + e \in \mathcal{I}\}$ if $f = s$. Then (e,f) is an edge of G if and only if $D(e,f) \neq \emptyset$, and its capacity $u(e,f)$ is defined to be $\sum(\lambda_i: i \in D(e,f))$. An attempt to understand how the capacities of edges of G are transformed by a crude augmentation helps to motivate some of the ideas we shall need.

An (r,s) dipath Q of G having vertex-sequence $r = e_0, e_1, \ldots, e_m, e_{m+1} = s$, is *shortcut-free* if for all i,j with $0 \leqslant i < j - 1 \leqslant m$, (e_i, e_j) is not an edge of G. A crude augmentation of amount δ on a shortcut-free path Q, lowers the capacity of each edge of Q by exactly δ. Moreover, if Q is still a dipath in G after the augmentation, it remains shortcut-free. (We do not present the proof

of these facts, but mention them only to motivate the following.) This suggests that repeated augmentation on the same path Q might have the eventual effect of a "grand" augmentation of amount equal to the minimum capacity of the edges of Q. However, it is not at all clear how many crude augmentions might be required, so we shall define the grand augmentation in a single step. Nevertheless, we need to understand the cumulative effect of successive crude augmentations in order to discover the grand augmentation.

We call the minimum capacity ε of the edges of an (r, s)-dipath Q, the *capacity* of Q. Just as the definition of a crude augmentation requires the choice of *one* (or a first) $i \in D(a, b)$ for each edge (a, b) of Q, the specification of successive augmentations requires an *order* for $D(a, b)$. To this end let us order J, say $J = \{1, 2, ..., |J|\}$ and define the *level* $L_i(a, b)$ of i in $D(a, b)$ to be $\sum (\lambda_j : j \in D(a, b), j \leqslant i)$. The motivation is the following. We picture the path Q with each edge (a, b) (other than the first one) having $|D(a, b)|$ blocks of capacity stacked on it. Each $i \in D(a, b)$ contributes one block of thickness λ_i, and the blocks are stacked in order of increasing i. Thus $L_i(a, b)$ is simply the height of (the top of) the block corresponding to I_i. The smallest stack has height ε (or more, if ε happens to be the capacity of the first edge of Q).

Let us suppose that one crude augmentation of amount δ has been performed, and imagine the new picture. (We are assuming that Q is shortcut-free.) All of the stacks have decreased in height by exactly δ. If I_i is the bottom block of the stack on (a, b), then a slab of thickness δ has been removed from that block. But other stacks may have had blocks corresponding to I_i. Now each such block will have a slab of thickness δ converted into a new block corresponding to a set $I_i + a - b$. It is convenient to choose the linear ordering of the new J so that this new block actually occupies the position of the converted slab. After a sequence of such augmentations, a block of initial capacity determined by I_i will have been subdivided a number of times. The part of it which has not been destroyed will now consist of new blocks of capacity corresponding to sets of the form $I_i' = (I_i \cup \{a_1, ..., a_k\}) \setminus \{b_1, ..., b_k\}$, where $(a_1, b_1), ..., (a_k, b_k)$ are edges of Q. (Note that possibly $k = 0$ or $b_i = s$ for some i.) We call these sets *mutations* of I_i. Part (or all, or none) of the original block could be destroyed because, for one or more of the augmentations, it, or a block corresponding to an earlier mutation, was the bottom one in its stack. This will occur if the sum of the amounts of crude augmentations exceeds $L_i(a, b) - \lambda_i$.

From the above ideas we conjecture that a grand augmentation might be constructed with the following property. If $i \in D(a, b)$ and $L_i(a, b) \geqslant \varepsilon + \lambda_i$ (so the whole block corresponding to I_i in the stack for (a, b) has height $\geqslant \varepsilon$), then every mutation of I_i contains b and not a. (Roughly, this block of capacity is completely unused.) If $L_i(a, b) \leqslant \varepsilon$ (so the whole block has height $\leqslant \varepsilon$), then every mutation of I_i contains a and not b. (This block is

completely consumed.) Finally, if $\varepsilon < L_i(a, b) < \lambda_i + \varepsilon$ (so part of the block has height exceeding ε and part has height less than ε), then there are some mutations of each type; those which contain b and not a have λ_i' which sum to $L_i(a, b) - \varepsilon$ and the others have λ_i' which sum to $\varepsilon + \lambda_i - L_i(a, b)$. In all cases the λ_i' for the mutations of I_i sum to λ_i.

Using the above ideas we now define formally the grand augmentation. For each $i \in J$ define B_i to be $\{(a, b)$: (a, b) an edge of Q, $i \in D(a, b)\}$. Order B_i as (a_0, b_0), (a_1, b_1),..., $(a_{q(i)-1}, b_{q(i)-1})$, such that $L_i(a_j, b_j) \leqslant L_i(a_{j+1}, b_{j+1})$ for $0 \leqslant j \leqslant q(i) - 2$. (There is some notational ambiguity here. We are actually defining a sequence for each $i \in J$, but the notation for the elements of the sequence does not reflect this dependence on i. However, the notation for the length, $q(i)$, of the sequence, does.) For convenience, define $L_i(a_{q(i)}, b_{q(i)})$ to be ∞.

For $j = 0, 1,..., q(i)$ define I_{ij} to be $(I_i \cup \{a_0,..., a_{j-1}\}) \setminus \{b_0, b_1,..., b_{j-1}\}$. Notice that $I_{i0} = I_i$. Also, the I_{ij} will have cardinality $|I_i| + 1$ or $|I_i|$, according to whether or not some b_l, $0 \leqslant l \leqslant j - 1$, is equal to s. We define λ_{ij} for $j = 0, 1,..., q(i)$ inductively, as follows:

$\lambda_{ij} = \max(0, \min(L_i(a_j, b_j) - \varepsilon, \lambda_i) - \sum (\lambda_{il}: l < j))$. We define y' to be $\sum (\lambda_{ij} I_{ij}: i \in J, 0 \leqslant j \leqslant q(i))$, and J' to be $\{(i, j): i \in J, 0 \leqslant j \leqslant q(i), \lambda_{ij} > 0\}$. The grand augmentation consists in replacing y by y', J by J', and $(\lambda_i: i \in J)$ by $(\lambda_{ij}: (i, j) \in J')$. We can now state the algorithm; it remains, of course, to prove that it is efficient (and valid).

ALGORITHM 4.1 (Input is a matroid M on E and a real-valued vector $x = (x_j: j \in E)$. We assume a fixed linear ordering of E.)

Step 1. If $x_j < 0$ for some j, stop; $x \notin P$.

Step 2. Put $J = \{0\}$, $I_0 = \varnothing$, $\lambda_0 = 1$, $y = 0$.

Step 3. If $x = y$, stop; $x \in P$.

Step 4. Form the digraph $G(J, \lambda)$. If there exists $A \subseteq E$ such that no edge of G leaves $A + r$, stop; $x(A) > r(A)$ and $x \notin P$.

Step 5. Find Q, the CBFS dipath in G from r to s. Perform a grand augmentation on Q, and go to Step 3.

As we have already observed, to find Q in Step 5 (or A in Step 4) one can use a standard graph-theoretic algorithm. The construction of G in Step 3 using an independence oracle can be accomplished in time polynomial in n, provided that $|J|$ is bounded by a polynomial in n. We shall prove that $|J|$ increases only moderately with each augmentation and so remains polynomially bounded provided the total number of augmentations is also polynomially bounded. This is one of the consequences of the next result.

(The reader may expect, from the definition of J', that after k augmen-

tations J consists of a set of $(k+1)$-tuples. This is not the intention. The definition of J' is stated in terms of pairs (i,j) only for convenience.)

LEMMA 4.2. *Consider a grand augmentation on the path Q with vertex-sequence $r, e_1, \ldots, e_m, s$. Then $\lambda_{ij} > 0$, $(i,j) \in J'$, and $\sum(\lambda_{ij}: (i,j) \in J') = 1$. Moreover, $y' = y + \varepsilon\{e_1\}$. Finally, $|J'| \leqslant |J| + m$.*

Proof. It is obvious from the definition of λ_{ij} that $\lambda_{ij} \geqslant 0$ for all (i,j), and that, for each $i \in J$, $\sum(\lambda_{ij}: 0 \leqslant j \leqslant q(i)) = \lambda_i$, so the λ_{ij} add to 1 because the λ_i do. Now for each $e \in E$, we define up(e) to be $\sum(\lambda_{ij}: e \in I_{ij}, e \notin I_i, i \in J, (i,j) \in J')$ and down(e) to be $\sum(\lambda_{ij}: e \notin I_{ij}, e \in I_i, i \in I, (i,j) \in J')$. Clearly, $y'_e = y_e + \text{up}(e) - \text{down}(e)$. If e is not a vertex of Q, then it is obvious that up$(e) =$ down$(e) = 0$, so $y'_e = y_e$. Now suppose that e is a vertex of Q, so that there is an edge (e,f) of Q. For each $i \in D(e,f)$ let p_i denote $\sum(\lambda_{ij}: e \in I_{ij}, (i,j) \in J')$; thus $\sum(p_i: i \in D(e,f)) = \text{up}(e)$. Now if $L_i(e,f) \leqslant \varepsilon$, then $p_i = \lambda_i$. If $L_i(e,f) > \varepsilon + \lambda_i$, then $p_i = 0$. Finally, if $\varepsilon < L_i(e,f) < \varepsilon + \lambda_i$ (this happens, of course, for at most one i), then $p_i = \varepsilon + \lambda_i - L_i(e,f)$. It follows that up$(e) = \sum(p_i: i \in D(e,f)) = \varepsilon$. A similar argument shows that, if there is an edge (d,e) of Q with $d \in E$ (i.e., $e \neq e_1$), then down$(e) = \varepsilon$, so $y'_e = y_e + \varepsilon - \varepsilon = y_e$. But of course down$(e_1) = 0$, since there is no (i,j) with $e_1 \in I_i$, $e_1 \notin I_{ij}$. Thus $y' = y + \varepsilon\{e_1\}$, as required.

To estimate $|J'|$, consider, for i fixed, $|\{j: 0 \leqslant j \leqslant q(i), \lambda_{ij} > 0\}| \leqslant 1 + |\{(e,f): (e,f) \in B_i, \varepsilon < L_i(e,f) < \varepsilon + \lambda_i\}|$. But each edge (e,f) of Q, $e \neq r$, can play this role for at most one i, so the desired bound is attained. ∎

We have yet to prove that the augmentation is valid, because we have not proved that the sets I_{ij} are independent. In addition, we want to understand how edges can enter and leave G as a result of a grand augmentation. The following technical results will deal with both of these matters. We remark that the first part of Lemma 4.3 is standard, having been used to validate matroid partition and intersection algorithms.

LEMMA 4.3. *Let e,f be distinct elements of E, let $I \in \mathscr{I}$, and let $a_0, b_0, a_1, b_1, \ldots, a_p, b_p$ be a sequence of distinct elements of E satisfying:*

(a) $a_i \notin I$, $b_i \in I$, $0 \leqslant i \leqslant p$;

(b) $b_i \in C(I, a_i)$, $0 \leqslant i \leqslant p$;

(c) $b_i \notin C(I, a_j)$, $0 \leqslant j < i \leqslant p$.

Then:

(i) $I' = (I \cup \{a_0, \ldots, a_p\}) \setminus \{b_0, \ldots, b_p\} \in \mathscr{I}$.

(ii) *If $e \notin I'$ and $I' + e \in \mathscr{I}$, then $e \notin I$ and $I + e \in \mathscr{I}$.*

(iii) *If* $f \in C(I', e)$ *and* $f \notin C(I, e)$, *then there exist* k, l, $0 \leqslant k \leqslant l \leqslant p$, *such that* $f \in C(I, a_k)$ *and* $(b_l = e$ *or* $b_l \in C(I, e))$.

Remark. If $f \in C(I', e)$ and $C(I, e)$ does not exist, we consider the hypothesis of (iii) to be satisfied.

Proof. The proof is by induction on p. First we treat the case $p = 0$. Then (i) is true by Lemma 2.1. For (ii) suppose that $e \notin I'$ and $I' + e \in \mathscr{I}$. Then $e \neq b_0$, so $e \notin I$. If $I + e \notin \mathscr{I}$, then it contains a circuit C containing e, and so $(I' + e) + b_0$ contains both C and $C(I, a_0)$, contradicting Lemma 2.1.

Now consider (iii) in the case $p = 0$. If $b_0 = e$, then $C(I', e) = C(I, a_0)$, so $f \in C(I, a_0)$, as required. If $b_0 \neq e$, then we know that $C(I, e)$ exists, as follows. Using (ii) with $p = 0$, and the fact that $C(I', b_0) = C(I, a_0)$, $I + e \in \mathscr{I}$ implies $e \in I$, so $e = b_0$. Now since $C(I, e) \neq C(I', e)$, we have $b_0 \in C(I, e)$. By Lemma 2.1, there is a circuit $C \subseteq C(I, e) \cup C(I, a_0)$ and not containing b_0. Again by Lemma 2.1, $C = C(I', e)$. But $f \in C(I', e) \setminus C(I, e)$, so $f \in C(I, a_0)$, as required.

Now suppose that $p \geqslant 1$ and that the statement of the lemma is true for all smaller values. We begin by showing that the set $I_1 = I + a_0 - b_0$ and the sequence $a_1, b_1, \ldots, a_p, b_p$ satisfy the hypotheses (a), (b), (c). That (a) still holds is clear. If (c) fails, then there exist i, j with $1 \leqslant j < i \leqslant p$ and $b_i \in C(I_1, a_j)$. By (iii) in the case $p = 0$, it follows that either $b_i \in C(I, a_j)$ which is not true, or that $b_i \in C(I, a_0)$ which is also not possible, and so (c) holds.

If (b) fails, then for some i, $1 \leqslant i \leqslant p$, we have $b_i \notin C(I_1, a_i)$. Again, applying (iii) with $p = 0$, we have $b_i \notin C(I, a_i)$, which is not true, or that $b_i \in C(I, a_0)$, which is also not possible, so (b) holds.

When we apply the inductive hypothesis to I_1 and the sequence a_1, $b_1, \ldots, a_p, b_p$, we obtain (i) instantly. We also have $e \notin I'$, $I' + e \in \mathscr{I}$ imply $e \notin I_1$, $I_1 + e \in \mathscr{I}$. Using (ii) for $p = 0$, the latter conditions imply $e \notin I$, $I + e \in \mathscr{I}$, and (ii) is proved.

For (iii), suppose that $f \in C(I', e)$, $f \notin C(I, e)$. If $f \in C(I_1, e)$, then from the $p = 0$ case, we get that $f \in C(I, a_0)$ and $e = b_0$ or $b_0 \in C(I, e)$, which gives the desired conclusion with $k = l = 0$. Otherwise, we obtain from the induction hypothesis that there exist k, l with $1 \leqslant k \leqslant l \leqslant p$, such that $a_k = f$ or $f \in C(I_1, a_k)$ and $b_l = e$ or $b_l \in C(I_1, e)$. Suppose that $f \notin C(I, a_k)$. Then $b_0 \in C(I, a_k)$, so there is a circuit $C \subseteq (C(I, a_k) \cup C(I, a_0)) - b_0$, by Lemma 2.1. Again by Lemma 2.1, we must have $C = C(I_1, a_k)$, so $f \in C$, and so $f \in C(I, a_0)$. It follows that we have $f \in C(I, a_k)$ or $f \in C(I, a_0)$. Now suppose that $b_l \neq e$ and $b_l \notin C(I, e)$. (By (ii) if $C(I, e)$ does not exist, then $e \in I$, so $e = b_0$; but then $b_l \in C(I_1, e) = C(I, a_0)$, contradicting condition (c).) It follows that $b_0 \in C(I, e)$ and hence $b_l \in C(I_1, e) \subseteq (C(I, e) \cup C(I, a_0)) - b_0$, so $b_l \in C(I, a_0)$, a contradiction. ∎

LEMMA 4.4. *Let e,f be distinct elements of E, let $I \in \mathscr{I}$, and let $a_0, b_0, ..., a_p, b_p, a_{p+1}$ be a sequence of distinct elements of E satisfying:*

(a) $a_i \notin I$, $b_i \in I$, $0 \leqslant i \leqslant p$, *and* $a_{p+1} \notin I$;

(b) $b_i \in C(I, a_i)$, $0 \leqslant i \leqslant p$;

(c) $b_i \notin C(I, a_j)$, $0 \leqslant j < i \leqslant p$;

(d) $I + a_{p+1} \in \mathscr{I}$.

Then:

(i) $I' = (I \cup \{a_0, ..., a_{p+1}\}) \setminus \{b_0, ..., b_p\} \in \mathscr{I}$.

(ii) *If $e \notin I'$ and $I' + e \in \mathscr{I}$, then $e \notin \mathscr{I}$ and $I + e \in \mathscr{I}$.*

(iii) *If $f \in C(I', e)$ and $f \notin C(I, e)$ then there exist k, l, $0 \leqslant k \leqslant l \leqslant p$, such that $f \in C(I, a_k)$ and ($b_l = e$ or $b_l \in C(I, e)$ or $e \notin I$ and $I + e \in \mathscr{I}$).*

Proof. The hypotheses of Lemma 4.3 are satisfied by $I + a_{p+1}$ and the sequence $a_0, b_0, ..., a_p, b_p$. The truth of (i) and (ii) are immediate consequences. For (iii), we obtain from Lemma 4.3 (iii) that there exist k, l, $0 \leqslant k \leqslant l \leqslant p$ such that $f \in C(I, a_k)$ and ($b_l = e$ or $b_l \in C(I + a_{p+1}, e)$). Suppose that $b_l \in C(I + a_{p+1}, e)$. If $I + e \notin \mathscr{I}$, then $C(I, e) = C(I + a_{p+1}, e)$, so $b_l \in C(I, e)$. If $I + e \in \mathscr{I}$, then $e \notin I$, since otherwise $(I + a_{p+1}) + e \in \mathscr{I}$, and (iii) is proved. ∎

It is important to observe that we may take any subsequence of the indices $0, 1, ..., p$ in Lemmas 4.3 or 4.4 and the hypotheses will be satisfied. Each I_{ij} in the definition of the grand augmentation is now seen to be independent, because, where the a_j, b_j are arranged in the order they appear on Q, the conditions of the lemmas are satisfied. This proves the validity of the grand augmentation. We proceed to prove that properties (I) and (II) of Theorem 3.1 are satisfied by the grand augmentation. For these purposes, we define an edge of a shortcut-free path Q of $G(J, \lambda)$ to be *critical* if it has minimum capacity among the edges of Q.

LEMMA 4.5. *Suppose that (J', λ') is obtained from (J, λ) by a grand augmentation on the shortcut-free path Q of $G(J, \lambda)$. Then any critical edge of Q is not an edge of $G(J', \lambda')$.*

Proof. Let (e,f) be a critical edge of Q. It is clear that the result is true if $e = r$, so suppose $e \in E$. Because (e,f) is critical, it follows that for every $i \in D(e,f)$ every mutation of I_i will contain e and not f, so no mutation of such I_i will cause (e,f) to be an edge of $G(J', \lambda')$. The only possibility is that, for some $i \notin D(e,f)$, a mutation I_{ij} of I_i satisfies $f \in C(I_{ij}, e)$ if $f \neq s$, or $e \notin I_{ij}$, $I_{ij} + e \in \mathscr{I}$ if $f = s$. In the first case it follows from Lemma 4.3 or Lemma 4.4 that there exist vertices a_k, b_l of Q with a_k preceding b_l on Q and

such that $f \in C(I_i, a_k)$ and ($b_l = e$ or $b_l \in C(I_i, e)$ or $e \notin I_i$ and $I_i + e \in \mathscr{I}$). Thus $f = a_k$ or (a_k, f) is an edge of G. It follows that $f = a_k$ or f precedes a_k on Q, because Q is shortcut-free. In addition, $b_l = e$ or one of (e, b_l) or (e, s) is an edge of G. It follows from this that e follows a_l on Q, because Q is shortcut-free. Therefore, f precedes e on Q, which is impossible. The second case, $e \notin I_{ij}$ and $I_{ij} + e \in \mathscr{I}$, is impossible by Lemma 4.3(ii) or 4.4(ii). ∎

LEMMA 4.6. *Suppose that (J', λ') is obtained from (J, λ) by a grand augmentation on the shortcut-free path Q of $G(J, \lambda)$. If (e, f) is an edge of $G(J', \lambda')$ but not an edge of $G(J, \lambda)$, then $e, f \in E$, and there exist vertices a, b of Q with a preceding b on Q such that $a = f$ or (a, f) is an edge of $G(J, \lambda)$ and $b = e$ or (e, b) is an edge of $G(J, \lambda)$.*

Proof. It is clear that $e = r$ is impossible. Moreover, $f = s$ is impossible by Lemma 4.3(ii) or 4.4(ii). Thus $e, f \in E$. Now using Lemma 4.3(iii) or 4.4(iii) we choose $a = a_k$ and $b = b_l$ or $b = s$. Then a precedes b on Q. Moreover, $a = f$ or (a, f) is an edge of $G(J, \lambda)$. Also $e = b$ or (e, b) is an edge of $G(J, \lambda)$. ∎

Lemmas 4.6, 4.7 verify that the sequence of digraphs and augmenting paths generated by Algorithm 4.1 satisfy the conditions of Theorem 3.1. Therefore, we obtain the following result on the complexity of the algorithm. (It is easy to check that it is true for $n = 0$ and 1.)

THEOREM 4.8. *Algorithm* 4.1 *terminates after at most n^3 augmentations.*

We can now prove Theorem 1.3. Apply the algorithm. There are at most n^3 augmentations, each of which increases $|J|$ by at most n. Since $|J|$ is initially 1, the final J has at most $n^4 + 1$ elements, as required. It is clear from the way in which the λ_i are modified by the algorithm that each λ_i is an integer combination of elements of $\{x_j : j \in E\} \cup \{1\}$. Algorithm 4.1 runs in polynomial time, assuming that we have a polynomial-time subroutine for recognizing independence in M. A more accurate estimate of the complexity would depend on the complexity of the independence oracle. Since counting the number of calls on the oracle is often misleading, it is more useful to estimate the running time for a concrete special class. Suppose that M is given by linear independence of columns of an r by n matrix N over a field, and let us count arithmetic operations in the field as elementary steps. Given an independent set I_i, row operations requiring $O(r^2 n)$ steps will transform N so that, for each $e \in E \setminus I_i$, it is easy to recognize whether $I_i + e \in \mathscr{I}$, and if not, to find $C(I_i, e)$. Thus there is an $O(r^2 n^5)$ algorithm to compute $G(J, \lambda)$ with its edge capacities. It is easy to see that finding and performing an augmentation requires much less work, so we have an $O(r^2 n^8)$ algorithm for this special case. Perhaps even less attractive is the $O(rn^4)$ space bound. Of

course, we would not expect these bounds to be approached in practice, and probably they can be lowered.

We do want to mention one variant of Algorithm 4.1 which has important advantages in practice. It entails using linear algebra to ensure that $|J| \leqslant n+1$ after each augmentation, for which we can use the standard method in linear programming for converting a feasible solution to a basic feasible solution. Thus we need to perform at most n pivot steps on a matrix of size at most $n+1$ by $2n+1$, which can be done in $O(n^3)$ time, counting real arithmetic operations as single steps. Since the new J will be a subset of the old, the properties of the auxiliary digraph needed to prove Theorem 4.8 are maintained. The disadvantages of this variant are that it requires nonadditive arithmetic, and that the discreteness properties of λ are lost. The advantages are a drastic decrease in space requirements, and possibly a considerable decrease in time requirements. When this approach is applied to the above matrix example, we obtain an $O(n^2)$ space bound and an $O(r^2n^5 + n^6)$ time bound.

5. Matroid Intersection Polyhedra

Suppose that we are given two matroids, $M_1 = (E, \mathscr{I}_1)$ and $M_2 = (E, \mathscr{I}_2)$ with respective rank functions r_1, r_2. Let P_k denote the convex hull of independent sets of M_k, $k = 1$ and 2, and let P denote the convex hull of common independent sets, that is, of members of $\mathscr{I}_1 \cap \mathscr{I}_2$. A famous theorem of Edmonds [8] says that $P = P_1 \cap P_2$. Therefore, to test a point x for membership in P, it is enough to apply Algorithm 4.1 twice, once to M_1 and x, once to M_2 and x. If $x \notin P$, we obtain a most violated constraint with respect to the description $\{y: y \geqslant 0,\ y(A) \leqslant \min(r_1(A), r_2(A)) \text{ for } A \subseteq E\}$ for P. Notice, however, that if $x \in P$ we do *not* obtain an explicit representation for x as a convex combination of common independent sets of M_1 and M_2.

Now let us define $r(A)$, for $A \subseteq E$, to be $\min(r_1(B) + r_2(A \setminus B)\colon B \subseteq A)$. (Edmonds also proved that $r(A)$ is the maximum cardinality of a common independent subset of A.) It is easy to see that another description of P is $\{y: y \geqslant 0,\ y(A) \leqslant r(A) \text{ for } A \subseteq E\}$. The problem of testing membership in P can thus be solved by minimizing $r(A) - x(A)$ over $A \subseteq E$. This is a different problem from the one that arises from the first description of P, and it is not solved by simply running Algorithm 4.1 twice. However, the following result [8] provides a good characterization of a most violated inequality. It also generalizes Theorem 1.2.

Theorem 5.1. *For any* $x = (x_e: e \in E) \geqslant 0$, $\max(y(E)\colon y \leqslant x,\ y \in P) = \min(r_1(A_1) + r_2(A_2) + x(E \setminus (A_1 \cup A_2))\colon A_1, A_2 \subseteq E)$.

The purpose of this section is to extend the methodology developed for Algorithm 4.1 to obtain a strongly polynomial algorithm for computing the optima of Theorem 5.1. We shall keep a feasible solution y to the maximization problem, as in the one-matroid case. However, keeping y explicitly as a convex combination of common independent sets of M_1, M_2 does not seem to work. (One reason for the difficulty is that no discreteness result like Theorem 1.3 is known in this case.) Rather we keep an expression for y as a convex combination of members of $\mathscr{I}_k$, for $k = 1$ *and* 2. Thus at a general step, we shall have $y = \sum(\lambda_i^1 I_i^1 : i \in J_1) = \sum(\lambda_i^2 I_i^2 : i \in J_2)$ where, for $k = 1$ and 2,

$$I_i^k \in \mathscr{I}_k \qquad \text{for} \quad i \in J_k;$$
$$\lambda_i^k > 0, \qquad i \in J_k;$$
$$\sum(\lambda_i^k : i \in J_k) = 1.$$

Initially, of course, we can take $J_1 = J_2 = \{0\}$, $\lambda_0^1 = \lambda_0^2 = 1$, $I_0^1 = I_0^2 = \emptyset$, $y = 0$.

We define a more complicated auxiliary digraph $G = G(J_1, J_2, \lambda^1, \lambda^2)$ as follows. (Roughly speaking, it is obtained by pasting together copies of the auxiliary digraphs determined by (M_1, J_1, λ^1) and (M_2, J_2, λ^2), one of which has all of its edges reversed.) Let $E_k = \{e^k : e \in E\}$ for $k = 1$ and 2. The vertex-set of G is $E_1 \cup E_2 \cup \{r, s\}$. With respect to (J_2, λ^2) and the matroid M_2, the set $D_2(e,f)$, for $e \in E$, $f \in E + s$, is defined as in Section 4. With respect to (J_1, λ^1) and matroid M_1, the set $D_1(e,f)$ for $e \in E$, $f \in E + r$ is defined as in Section 4 with s replaced by r. For each $e \in E$, (e^1, e^2) is an edge of G if and only if $y_e < x_e$, and $u(e^1, e^2) = x_e - y_e$. Similarly (e^2, e^1) is an edge of G and if only if $y_e > 0$, and $u(e^2, e^1) = y_e$. For distinct elements e, f of E, (e^2, f^2) is an edge of G if and only if $D_2(e,f) \neq \emptyset$, and $u(e^2, f^2) = \sum(\lambda_i^2 : i \in D_2(e,f))$; (f^1, e^1) is an edge of G if and only if $D_1(e,f) \neq \emptyset$, and $u(f^1, e^1) = \sum(\lambda_i^1 : i \in D_1(e,f))$. For $e \in E$, (e^2, s) is an edge of G if and only if $D_2(e,s) \neq \emptyset$, and $u(e^2, s) = \sum(\lambda_i^2 : i \in D_2(e,s))$; (r, e^1) is an edge of G if and only if $D_1(e,r) \neq \emptyset$, and $u(r, e^1) = \sum(\lambda_i^1 : i \in D_1(e,r))$. We have the following augmenting path theorem, from which Theorem 5.1 follows. Notice that, in contrast to Theorem 2.2, here we cannot insist that components of y are never lowered.

THEOREM 5.2. *If there is a directed path from r to s in G, then there exists $y' \in P$, $y' \leqslant x$, such that $y'(E) > y(E)$. If there is no such path, then there exist sets $A_1, A_2 \subseteq E$ such that $y(E) = r_1(A_1) + r_2(A_2) + x(E \setminus (A_1 \cup A_2))$.*

Proof. First, suppose that no path exists. Then there is a set $B \subseteq E_1 \cup E_2$

such that no edge of G leaves $B+r$. Let $A_1=\{e: e\in E, e^1\notin B\}$ and $A_2=\{e: e\in E, e^2\in B\}$. Since there is no edge of the type (e^2,f^2) with $e\in A_2$, $f\notin A_2$, and none of the type (e^2,s) with $e\in A_2$, we conclude, as in the proof of Theorem 2.2, that $y(A_2)=r_2(A_2)$. Since there is no edge of the type $((f^1,e^1)$ with $e\in A_1$, $f\notin A_1$ or of the type (r,e^1) with $e\in A_1$, we conclude that $y(A_1)=r_1(A_1)$. Since there is no edge of the form (e^1,e^2) with $e\notin A_1$, $e\notin A_2$, we conclude that $y_e=x_e$ for all $e\in E\setminus(A_1\cup A_2)$. Finally, since there is no edge of the type (e_2,e_1) with $e\in A_2$, $e\in A_1$, we conclude that $y_e=0$ for all $e\in A_1\cap A_2$. Hence

$$\begin{aligned} y(E) &= y(A_1)+y(A_2)+y(E\setminus(A_1\cup A_2)) \\ &= r_1(A_1)+r_2(A_2)+x(E\setminus(A_1\cup A_2)) \qquad \text{as required.} \end{aligned}$$

Now suppose that there is a directed path Q from r to s in G, and let $\delta>0$ be sufficiently small. For each edge (e^1,e^2) of Q, let $y'_e=y_e+\delta$. For each edge (e^2,e^1) of Q let $y'_e=y_e-\delta$. Otherwise let $y'_e=y_e$. For each edge (e^2,f^2) (or (e^2,s)) of Q, choose $i\in D_2(e,f)$ (or $i\in D_2(e,s)$). Introduce the set I_i^2+e-f (or I_i^2+e) with coefficient δ. For each chosen i, let k_i^2 denote the number of such edges of Q for which i is chosen. Replace λ_i^2 by $\lambda_i^2-k_i^2\delta$. Apply a similar construction to the edges of Q of the form (f^1,e^1) or (r,e^1). It is straightforward to check that the family $(I_i^k: i\in J_k)$ and vector $(\lambda_i^k: i\in J_k)$ provide an expression for y' as a convex combination of independent sets of M_k, $k=1$ and 2. Moreover, $y'(E)=y(E)+\delta>y(E)$, as required. ∎

We want to define a grand augmentation for a shortcut-free (r,s) dipath Q in G, which produces $y'\in P$ with $y'(E)=y(E)+\varepsilon$, where ε is the minimum capacity of edges of Q. We fix a linear ordering of J_2 and, for each edge of Q of the form (e^2,f^2) (or (e^2,s)) and each $i\in D_2(e,f)$ (or $D_2(e,s)$), we define the level $L_i^2(e,f)$ (or $L_i^2(e,s)$) as in Section 4. Similarly we define a new convex combination of members of $\mathscr{I}_2$. The sets I_{ij}^2 formed from an initial set I_i^2 satisfy the hypotheses of Lemma 4.3 or 4.4, and so they *are* independent in M_2. To show that the new J_2,λ^2 yield an expression for y', one needs a straightforward extension of the argument of Lemma 4.2. One also easily checks that $|J_2|$ has increased in size by at most the number of edges of Q having both ends in E_2+s. A similar method is used to update J_1 and λ^1, and analogous results hold.

We want to show that the algorithm obtained by fixing a linear ordering of $E_1\cup E_2$ and successively performing grand augmentations on CBFS (r,s)-dipaths is strongly polynomial. The methods used in Section 4 are easily adapted to show that a critical (minimum capacity) edge of Q of the form (r,e^1), (e^2,s), (f^1,e^1), or (e^2,f^2) is no longer an edge of G after the augmentation. Similarly an edge of the form (r,e^1) or (e^2,s) cannot appear

in G as the result of an augmentation. Finally, if an edge of the type (f^1, e^1) or (e^2, f^2) appears, then vertices a, b of Q exist, as in (II) of Theorem 3.1. To apply Theorem 3.1 it remains only to deal with the edges of G of type (e^1, e^2) and (e^2, e^1). But it is obvious that a critical edge of Q of either of these types will disappear after the augmentation on Q. Moreover, an edge (e^1, e^2) (resp. (e^2, e^1)) appears after augmentation on Q only if (e^2, e^1) (resp. (e^1, e^2)) was an edge of Q, so (II) is verified for such edges. It follows that the resulting algorithm will terminate after at most $(2n)^3$ augmentations. In fact, except for constant factors, it has the same time bound as Algorithm 4.1.

In the remainder of this section we describe a variant of the above algorithm and its connection with other matroid intersection polyhedra. We assume for convenience that $r(E_1) = r(E_2) = r$ (say), and we denote the convex hull of common bases of M_1, M_2 by P'. The algorithmic problem we will solve is: Given vectors l, u with $0 \leqslant l \leqslant u$, find if possible $y \in P'$ with $l \leqslant y \leqslant u$. We can solve this problem by maximizing $y(E)$ over $\{y: y \in P, l \leqslant y \leqslant u\}$; we have a solution if and only if the maximizing y satisfies $y(E) = r$.

We solve the maximization problem in two phases. In the first phase we apply the algorithm of this section to M_1, M_2 and $x = l$. It either provides expressions $\sum(\lambda_i^1 I_i^1 : i \in J_1) = \sum(\lambda_i^2 I_i^2 : i \in J_2)$ for l, or its concludes that $l \notin P$. In the latter case, of course, $\{y: l \leqslant y \leqslant u, y \in P\} = \varnothing$, and we are done, since $P' \subseteq P$. Otherwise, we again apply the algorithm of this section with $x = u$ and with the initial $y = l$, but with a slight twist. Namely, we now enforce a lower bound of l on y, by redefining the capacity of an edge of the form (e^2, e^1) to be $y_e - l_e$ (and including such an edge only if that capacity is positive).

The resulting algorithm has the properties required to apply Theorem 3.1, and so terminates after $O(n^3)$ augmentations. Clearly, at termination $y \in P$ and y satisfies $l \leqslant y \leqslant u$. If $y(E) = r$, then $y \in P'$, as required. Otherwise, by essentially the same argument as in the proof of Theorem 5.1, we have $A_1, A_2 \subseteq E$ with $y(A_1) = r_1(A_1)$, $y(A_2) = r_2(A_2)$, $y(E \setminus (A_1 \cup A_2)) = u(E \setminus (A_1 \cup A_2))$, and $y(A_1 \cap A_2) = l(A_1 \cap A_2)$. Thus we have

$$\begin{aligned} r > y(E) &= y(A_1) + y(A_2) - y(A_1 \cap A_2) + y(E \setminus (A_1 \cup A_2)) \\ &= r_1(A_1) + r_2(A_2) - l(A_1 \cap A_2) + u(E \setminus (A_1 \cup A_2)). \end{aligned}$$

This leads to the following characterization, due to McDiarmid [17].

THEOREM 5.3. *There exists a convex combination y of common bases satisfying $l \leqslant y \leqslant u$ if and only if for all $A_1, A_2 \subseteq E$ we have*

$$r_1(A_1) + r_2(A_2) + u(E \setminus (A_1 \cup A_2)) \geqslant r + l(A_1 \cap A_2).$$

Proof. Suppose no such y exists. If $l \notin P$, then we may suppose that there exists $A_1 \subseteq E$ with $l(A_1) > r_1(A_1)$. (The other case is symmetric.) Choose $A_2 = E$. Since $r_2(A_2) = r$ and $u(E \setminus (A_1 \cup A_2)) = 0$, the condition is violated. On the other hand if $l \in P$, then the condition is violated, by the argument preceding the statement of the theorem.

Now suppose that such y exists. Then for any $A_1, A_2 \subseteq E$, we have

$$\begin{aligned} r = y(E) &= y(A_1) + y(A_2) - y(A_1 \cap A_2) + y(E \setminus (A_1 \cup A_2)) \\ &\leqslant r_1(A_1) + r_2(A_2) - l(A_1 \cap A_2) + u(E \setminus (A_1 \cup A_2)), \end{aligned}$$

as required. ∎

The following consequences derived independently in [3, 9, 17] and conjectured by Fulkerson [12], give linear descriptions of the blocking and anti-blocking polyhedra associated with the common bases of two matroids. The methods of this section give strongly polynomial membership algorithms for these polyhedra.

COROLLARY 5.4. *The vector sum of the convex hull of common bases with the nonnegative orthant is* $\{x: x \geqslant 0,\ x(E \setminus (A_1 \cup A_2)) \geqslant r - r_1(A_1) - r_2(A_2)$ *for* $A_1, A_2 \subseteq E\}$.

COROLLARY 5.5. *The convex hull of subsets of common bases is* $\{x: x \geqslant 0,\ x(A_1 \cap A_2) \leqslant r_1(A_1) + r_2(A_2) - r$ *for* $A_1 \cup A_2 = E\}$.

6. SUBMODULAR FUNCTION MINIMIZATION

Suppose that we are given a function g on subsets of E which is submodular, that is, $g(A) + g(B) \geqslant g(A \cup B) + g(A \cap B)$ for all $A, B \subseteq E$. An important problem is to find $A \subseteq E$ such that $g(A)$ is minimum. A classical special case occurs when g is a cut capacity function: For a digraph having vertex-set $E \cup \{r, s\}$ and positive edge weights u_j, $g(A) = \sum (u_j: j$ leaves $A + r)$. Then minimizing g is the usual network minimum cut problem, for which there is a well-known strongly polynomial algorithm [6, 10]. Another choice for g is: For a matroid M on E having rank function r, and a real vector $(x_j: j \in E)$, $g(A) = r(A) + x(E \setminus A)$. Thus the main result of this paper is a strongly polynomial algorithm for submodular function minimization in a particular case.

For the general problem we usually assume that g is available only via an oracle which, given A, evaluates $g(A)$. One then would count a call on the oracle as a single step of the solution algorithm. Using the ellipsoid method

Grötschel *et al.* [13] give an algorithm for minimizing g which runs in time polynomial in n and the logarithm of $\max|g(A)|$. (This latter factor is necessary, even if one counts arithmetic operations as single steps.) It is an important open problem to find a strongly polynomial algorithm for submodular function minimization, or even to find a practical combinatorial method which runs in polynomial time.

If we define x_j to be $g(E-j)-g(E)$ for each $j \in E$, and define a function f by $f(A)=g(A)+x(A)-g(\varnothing)$, then to minimize g it is enough to minimize $f(A)-x(A)$. The advantage of this trick is that f, like a matroid rank function, is submodular, increasing ($f(A) \geqslant f(B)$ if $A \supseteq B$), and normalized ($f(\varnothing)=0$). Such a function is sometimes called a *polymatroid* function, and the polyhedron $P=\{y: y \geqslant 0,\ y(A) \leqslant f(A) \text{ for all } A \subseteq E\}$ is a *polymatroid*. Since an oracle for g obviously yields x and an oracle for f, the possibility of minimizing g by a generalization of Algorithm 4.1 suggests itself. Some partial results in this direction can be summarized. Theorem 1.2 is still true when r is replaced by f [8]. (Notice that we may assume that $x \geqslant 0$; any j for which $x_j < 0$ cannot be an element of a minimizing A.) Moreover, it does provide a good characterization of the minimum, because the maximizing y can be expressed as a convex combination of a small number of vertices of P, and these can be generated efficiently by the "polymatroid greedy algorithm" [8]. In [2] essentially all of the results of Section 2 are generalized to polymatroids, and a polynomial-time algorithm for constructing the auxiliary digraph is given. However, to date no analog of the refinement described in Section 4 is known.

It is not completely obvious that the two "well-solved" instances of submodular function minimization mentioned above, are actually well solved in the oracle context. For example, suppose that we wish to minimize g, which is available via an evaluation oracle, and we know that g is the cut capacity function of some weighted digraph, but we are not given the digraph. Does there exist a strongly polynomial algorithm? An obvious approach is to attempt to use the oracle to construct efficiently an appropriate digraph and then solve a minimum cut problem. This can be done; details are given in [4], which also treats some other instances of submodular function minimization related to minimum cut. We wish to show that Algorithm 4.1 similarly provides a strongly polynomial oracle algorithm for minimizing functions g given by $g(A)=r(A)-x(A)$, where r is the rank function of an unknown matroid and x is an unknown vector. If for some $j \in E$, $g(\{j\})+g(E-j)=g(E)$, then it is easy to see that $g(\{j\})+g(A)=g(A+j)$ for all $A \subseteq E-j$. Thus j must be included in any minimizer if $g(\{j\})<0$, and otherwise j can be excluded. After dealing with all such elements, we shall have reduced the problem to one on subsets of E', say, in which $g(\{j\})+g(E'-j)>g(E')$ for all $j \in E'$. But this is equivalent to $r(\{j\})+r(E'-j)>r(E')$, which implies that $r(\{j\})=1$, and hence that

$x_j = 1 - g(\{j\})$ for all $j \in E'$. Thus x is determined, and so we have an oracle to evaluate r and Algorithm 4.1 applies.

In the remainder of this section we describe extensions of Algorithm 4.1 to two more general problems of submodular function minimization. However, unlike the minimum cut and matroid cases, we are unable to handle these applications in a pure oracle setting. In each of them the function to be minimized arises from a combination of matroids and/or graphs, but the solution method needs to use these structures. Because of the similarity of the resulting algorithms to those of the previous two sections, we omit most of the details. It should be enough to indicate what structure the algorithm must maintain, and to define the auxiliary digraph.

The first extension is to polymatroid functions which are positive combinations of matroid rank functions. Suppose we are given a matroid $M_k = (E, \mathscr{I}_k)$ having rank function r_k and a number $a_k > 0$, for $1 \leqslant k \leqslant m$. We wish to minimize $f(A) - x(A)$ over $A \subseteq E$, where x is a (nonnegative) vector and $f(A) = \sum (a_k r_k(A): 1 \leqslant k \leqslant m)$. Algorithm 4.1, of course, solves the case $m = 1$, $a_1 = 1$. The related maximization problem is to maximize $y(E)$ over $y \in P$, where $P = \{y: y \geqslant 0,\ y(A) \leqslant f(A) \text{ for } A \subseteq E\}$. An important fact, which our algorithm proves, is that P is the vector sum of the polyhedra $P_k = \{y: y \geqslant 0,\ y(A) \leqslant a_k r_k(A) \text{ for } A \subseteq E\}$. Obviously, the vertices of P_k are the vectors of the form $a_k I$, $I \in \mathscr{I}_k$. So an equivalent statement is that P is the convex hull of all the vectors of the form $\sum (a_k I^k: 1 \leqslant k \leqslant m)$, where $I^k \in \mathscr{I}_k$ but, unlike the case $m = 1$, not all such vectors are vertices of P. As in the one-matroid case, the algorithm maintains $y \in P$ with $y \leqslant x$ and an explicit representation

$$y = \sum \left(a_k \sum (\lambda_i^k I_i^k: i \in J_k): 1 \leqslant k \leqslant m \right) \qquad \text{where}$$

$$I_i^k \in \mathscr{I}_k, \qquad i \in J_k, \qquad 1 \leqslant k \leqslant m;$$

$$\lambda_i^k > 0, \qquad i \in J_k, \qquad 1 \leqslant k \leqslant m;$$

$$\sum (\lambda_i^k: i \in J_k) = 1, \qquad 1 \leqslant k \leqslant m.$$

The auxiliary digraph $G = G(J, \lambda)$ has vertex-set $E \cup \{r, s\}$. For distinct elements e, f with $e \in E$ and $f \in E + s$, we define $D(e,f)$ to be $\{(i,k): 1 \leqslant k \leqslant m,\ i \in J_k,\ I_i^k + e \notin \mathscr{I}_k,\ f \in C(I_k^i, e)\}$ if $f \neq s$, and to be $\{(i,k): 1 \leqslant k \leqslant m,\ i \in J_k,\ e \notin I_i^k,\ I_i^k + e \in \mathscr{I}_k\}$ if $f = s$. Then (e,f) is an edge of G provided $D(e,f) \neq \varnothing$, and $u(e,f)$ is $\sum (a_k \lambda_i^k: (i,k) \in D(e,f))$. As usual, for $e \in E$, (r,e) is an edge of G if $y_e < x_e$, and $u(r,e) = x_e - y_e$. The time and space bounds for the resulting algorithm will be on the order of m times those for Algorithm 4.1.

As a second application, suppose that we are given a bipartite graph H having bipartition $\{E, F\}$, and a matroid M on F having rank function r. For

$A \subseteq E$, $N(A)$ denotes the set of vertices in F adjacent to at least one vertex of A. Then f, defined on subsets A of E by $f(A) = r(N(A))$, is a polymatroid function. Again, given a vector $x \geqslant 0$, we wish to minimize $f(A) - x(A)$. The special case in which H consists of disjoint edges is essentially the problem solved by Algorithm 4.1. The special case of the previous application, where $f(A) = \sum (r_k(A)\colon 1 \leqslant k \leqslant m)$, is also a special case of the present one: Make a copy of matroid M_k on $E_k = \{e^k\colon e \in E\}$, let $F = \bigcup (E_k\colon 1 \leqslant k \leqslant m)$, let M be the direct sum of these (disjoint) copies of M_k, and form H by joining each $e \in E$ to every copy e^k of e.

In this application there is also a useful characterization of $P = \{z\colon z \geqslant 0,$ $z(A) \leqslant f(A)$ for $A \subseteq E\}$. Namely, $z \in P$ if and only if there exists $y \in P'$, the convex hull of independent sets of M, and a "flow" $w = (w_{ef}\colon e \in E, f \in F, ef$ an edge of $H) \geqslant 0$ with the following properties. First, $\sum (w_{ef}\colon f \in F, ef$ an edge of $H) = z_e$ for all $e \in E$; second, $\sum (w_{ef}\colon e \in E, ef$ an edge of $H) = y_f$, for all $f \in F$. It is obvious that this condition is sufficient for z to be in P, for

$$\begin{aligned} z(A) &= \sum (w_{ef}\colon e \in A, f \in F, ef \text{ an edge of } H) \\ &\leqslant \sum (w_{ef}\colon e \in E, f \in N(A), ef \text{ an edge of } H) \\ &= y(N(A)) \\ &\leqslant r(N(A)) \\ &= f(A) \qquad \text{for} \quad A \subseteq E. \end{aligned}$$

That it is also necessary can be deduced from the algorithm we shall outline.

The above characterization suggests a flow approach, keeping z, w, and y, with an explicit expression $\sum (\lambda_i I_i\colon i \in J)$ for y as a convex combination of independent sets of M. With respect to w, J, λ, we define the auxiliary digraph G as follows. Its vertex-set is $E \cup F \cup \{r, s\}$. There is an edge (r, e) for each $e \in E$ for which $z_e < x_e$, and its capacity is $x_e - z_e$. There is an edge (e, f) for every $e \in E$ and every $f \in F$ for which ef is an edge of H, and its capacity is ∞. There is an edge (f, e) for every $e \in E$ and $f \in F$ such that $w_{ef} > 0$, and its capacity is w_{ef}. Where $D(e,f)$ for $e \in F$ and $f \in F + s$ is defined as in Section 4, there is an edge (e,f) for every e, f such that $D(e,f) \neq \emptyset$, and its capacity is $\sum (\lambda_i\colon i \in D(e,f))$.

Applying the same methods as before, we get an algorithm for minimizing $f(A) - x(A)$ which terminates after at most $(|E| + |F|)^3$ augmentations and hence is strongly polynomial (with respect to input size $|E| + |E(H)|$). Notice that an analogue of Theorem 1.3 can also be proved on the discreteness of λ and w.

The algorithms we have already described can be combined and extended in various ways. For example, in the last application one can allow H to be any digraph with given positive edge-capacities $(u_j\colon j \in E(H))$, let E, F be

any subsets of $V(H)$, M be a matroid on F, and define f by $f(A) = \min(\sum (u_j: j \text{ leaves } Q) + r(F \cap Q): Q \supseteq A)$, for $A \subseteq E$. One can also combine the two applications of this section, replacing r by $\sum a_k r_k$, or allow the more general polymatroids of this section in the intersection format of the last. One such extension, combining a capacitated bipartite graph with a nonnegative combination of (rank-one) matroid rank functions, could be used to give an algorithm for a pre-emptive scheduling problem solved by Martel [16]. (However, Martel's algorithm is more efficient than one obtained by a straightforward application of the present techniques.)

7. Some Remarks

Improving Theorem 1.3

A special case of Theorem 1.3 says that if x is rational-valued and D is a common denominator for the components of x, then a maximizing y can be chosen so that $|J| \leqslant n^4 + 1$ and $D\lambda$ is integer-valued. If we are given a nonnegative integer-valued vector b and are asked to find a family $(I_i: i \in J)$ of independent sets such that $b_e = |\{i: e \in I_i\}|$ for each $e \in E$, then the least possible $|J|$ is the least integer k such that $b/k \in P$, that is, it is $\max(\lceil b(A)/r(A) \rceil: \varnothing \neq A \subseteq E)$. This follows from Theorem 1.1 and this rational version of Theorem 1.3, but it is also an easy consequence of the matroid partition theorem [7]. However, this result is strengthened by Theorem 1.3, which provides a polynomial bound on the number of *distinct* sets I_i needed. Motivated by this last result, Schrijver [21] proved the following theorem, greatly improving this polynomial bound.

THEOREM 7.1. *Let $b = (b_e: e \in E) \geqslant 0$ be integer-valued. There is a family $(I_i: i \in J)$ of independent sets having $b = \sum (\lambda_i I_i: i \in J)$, having $\sum (\lambda_i: i \in J)$ a minimum, and having at most $2n$ distinct members.*

It is quite easy to prove from Theorem 7.1 that, when x is rational-valued, $n^4 + 1$ can be replaced by $2n + 1$. One can make the same improvement in Theorem 1.3 itself by an argument using rational approximation. Even $2n + 1$ is probably not near to being best possible here; I know of no lower bound better then the obvious $n + 1$. Schrijver's proof of Theorem 7.1 uses a technique introduced in [14]. The basic idea is that an optimal solution to the integer program minimize $(1 \cdot z: Az = b, z \geqslant 0, \text{integer-valued})$, where the columns of A are the incidence vectors of independent sets, may be constructed in a special way. More specifically, such a solution may be obtained by rounding down an optimal solution to the corresponding linear program, and then adding an optimal integer solution to a similar problem having "smaller" b. It would be nice to have a version of Algorithm 4.1

which maintained the discreteness properties of the λ_i as well as a linearly bounded $|J|$.

Matroid Polyhedra from Graphs

In some special cases there are simpler methods than Algorithm 4.1 to test a given x for membership in the polyhedron of a matroid M. For several classes of matroids arising from graphs, if one knows the relevant graph, the membership problem can be solved using minimum cut calculations. (See [4].) These classes include the classical forest matroids, for which the solution comes from an idea of Picard and Queyranne [19], and has been discovered independently by Padberg and Wolsey [18].

Polymatroids

An interesting consequence of Algorithm 4.1, and also a well-known fact, is that any *maximal* y satisfying $y \leqslant x$ and $y \in P$, also maximizes $y(E)$. Thus the polyhedron $P' = \{y: y \leqslant x, y \in P\}$ is a polymatroid. Moreover, we can also use the algorithm to answer the following basic question about P':

(*) Given $y \in P'$ and $e \in E$, find the largest ε such that $y + \varepsilon\{e\} \in P'$.

(Namely, run the algorithm with $x_j = y_j$, $j \neq e$, and $x_e = 1$.) There are two interesting algorithmic consequences of this fact. First, consider the optimization problem:

(**) maximize $(c \cdot y: y \leqslant x, y \in P)$

of which Algorithm 4.1 solves the special case in which every component of c is 1. Since P' is a polymatroid, we can apply the polymatroid greedy algorithm [8] to solve (**), using Algorithm 4.1 as a subroutine to answer (*).

Second, consider the problem (solved in Section 5):

(***) maximize $(y(E): y \leqslant x,\ y \in P_1 \cap P_2)$,

where P_1 and P_2 are matroid polyhedra. Clearly, this problem is equivalent to: maximize $(y(E): y \in P_1' \cap P_2')$, where P_1', P_2' are polymatroids as above. Schönsleben [20] and Lawler and Martel [15] have given a strongly polynomial algorithm for this latter problem for *any* polymatroids P_1', P_2', *assuming* the availability of oracles which can answer (*) in each of P_1', P_2'. (For a general polymatroid P', however, (*) is equivalent to submodular function minimization.) So we can apply one of these algorithms, using Algorithm 4.1 as a subroutine, to obtain a strongly polynomial algorithm for (***). The resulting algorithm appears to require $O(n^5)$ applications of Algorithm 4.1, however, so it is much less efficient than the algorithm of Section 5. Finally, we mention that, under the same oracle assumption, the

weighted polymatroid intersection problem $\max(c \cdot y: y \in P_1' \cap P_2')$ can be solved in polynomial time [5]. Thus (***) with $y(E)$ replaced by $c \cdot y$ can be solved in polynomial time, using Algorithm 4.1 as a subroutine.

ACKNOWLEDGMENTS

I wish to acknowledge conversations with Bob Bixby, Lex Schrijver, and András Frank. In particular, Frank suggested the final, crucial refinement of the algorithm: the lexicographic technique. I am also grateful to Bixby and Uri Peled for their detailed comments on an earlier version of the paper.

REFERENCES

1. R. E. BIXBY, Private communication, 1980.
2. R. E. BIXBY, W. H. CUNNINGHAM, AND D. M. TOPKIS, The poset of a polymatroid extreme point, *Math. Oper. Res.*, in press.
3. W. H. CUNNINGHAM, An unbounded matroid intersection polyhedron, *Linear Algebra Appl.* **16** (1977), 209–215.
4. W. H. CUNNINGHAM, Minimum cuts, modular functions, and matroid polyhedra, *Networks*, in press.
5. W. H. CUNNINGHAM AND A. FRANK, A primal dual algorithm for submodular flows, *Math. Oper. Res.*, in press.
6. E. DINIC, Algorithm for solution of a problem of maximum flow in a network with power estimation, *Soviet Math. Dokl.* **11** (1970), 1277–1280.
7. J. EDMONDS, Minimum partition of a matroid into independent sets, *J. Res. Nat. Bur. Standards* **69B** (1965), 67–72.
8. J. EDMONDS, Submodular functions, matroids, and certain polyhedra, *in* "Combinatorial Structures" (R. K. Guy *et al.*,), pp. 69–87, Gordon & Breach, New York, 1970.
9. J. EDMONDS AND R. GILES, A minmax relation for submodular functions on directed graphs, *Ann. Discrete Math.* **1** (1977), 185–204.
10. J. EDMONDS AND R. M. KARP, Theoretical improvements in algorithmic efficiency for network flow problems, *J. Assoc. Comput. Math.* **19** (1972), 248–264.
11. A. FRANK, Finding feasible vectors of Edmonds–Giles polyhedra, *J. Combin. Theory Ser. B*, in press.
12. D. R. FULKERSON, Blocking and anti-blocking pairs of polyhedra, *Math. Programming* **1** (1971), 168–194.
13. M. GRÖTSCHEL, L. LOVÁSZ, AND A. SCHRIJVER, The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* **1** (1981), 169–197.
14. M. GRÖTSCHEL, L. LOVÁSZ, AND A. SCHRIJVER, "Polynomial Algorithms for Perfect Graphs," Report No. 81176-OR, Universität Bonn, 1981.
15. E. L. LAWLER AND C. MARTEL, Computing maximal polymatroidal network flows, *Math. Oper. Res.* **7** (1982), 334–347.
16. C. MARTEL, Preemptive scheduling with release times, deadlines, and due times, *J. Assoc. Comput. Mach.* **29** (1982), 812–829.
17. C. J. H. MCDIARMID, Blocking, antiblocking, and pairs of matroids and polymatroids, *J. Combin. Theory Ser. B* **25** (1978), 313–325.

18. M. PADBERG AND L. WOLSEY, "Trees and Cuts," C.O.R.E. Discussion Paper 8138, Louvain, 1981.
19. J. C. PICARD AND M. QUEYRANNE, Selected applications of minimum cuts in networks, *INFOR–Canada J. Oper. Res. Inform. Process.* **20** (1982), 394–422.
20. P. SCHÖNSLEBEN, "Ganzzahlige Polymatroid-Intersektions-Algorithmen," Ph.D. dissertation ETH, Zurich, 1980.
21. A. SCHRIJVER, Private communication, 1981.
22. D. J. A. WELSH, "Matroid Theory," Academic Press, New York, 1976.
23. L. WOLSEY, Private communication, 1981.